

# ソフトウェア ドキュメンテーション

*Software Documentation*

# まえがき

ソフトウェアを開発していく上で、ドキュメンテーションはどんな意味を持っているのでしょうか？ そしてソフトウェアドキュメントにはどんなものがあり、それぞれどんなことを記述すればよいのでしょうか？

本書の目的は、1つのソフトウェアを開発していく上で、ドキュメンテーションがどんな意味を持つかを明らかにし、開発を進めるためにどのようにドキュメントを作成していけばよいかの指針を示すことです。

本書では、ソフトウェアを開発するため、またソフトウェアを使用するために必要となるドキュメントを、ソフトウェアドキュメントと呼びます。そして、ソフトウェアドキュメントの種類・役割・内容を、各ドキュメントを作成する立場から整理してまとめました。特に、本書はソフトウェア開発者を対象としています。本書では、開発用ドキュメンテーションはもちろん、開発者がソフトウェアのオンラインヘルプやマニュアルなどを作成する場合も想定して、開発者のためのユーザー向けドキュメンテーションに関する基本的なことも解説しています。そして、どちらを作成する上でも重要となるテクニカルライティング技術としての注意事項を「ドキュメンテーションの作法」として紹介します。

第一部「開発者のためのドキュメンテーション」では、ソフトウェアの開発用ドキュメントとは何か、その役割と記述の指針を述べ、技術的なポイントを紹介します。

第2章以降では、ソフトウェアの開発工程の中での、それぞれのドキュメントの位置づけ、役割を述べます。各章のはじめには、その章で述べるドキュメントが、開発工程の中でどこに位置するかを図で示しています。

各ドキュメントの記述項目の節では、ほぼ網羅的に項目を挙げています。しかし、実際のドキュメンテーションの場合は、対象となるプログラムやシステムによって、本書の記述項目が全て必要であるとは限りません。その場合には、本書で挙げている記述項目を参考として、対象となるソフトウェアに適した項目を抽出し、適した目次を構成していただきます。また、特殊なソフトウェアでは、本書で挙げたもののほかに特別な記述項目が必要となる場合もあります。

第二部「ユーザーのためのドキュメンテーション」では、ソフトウェアを使用するユーザーに提供するドキュメントを作成する上での指針を記述します。ユーザーの役に立つソフトウェアドキュメンテーションとは何か、どのようなことを記述すればユーザーの助けになるかを考えていきます。また、ユーザーのためのソフトウェアドキュメントにおける媒体（紙や電子媒体）の違いによるそれらの役割や作成上の注意点についても考察しています。

第三部「ドキュメンテーションの作法」では、第一部や第二部で紹介したドキュメントの品質を上げるための一般的なテクニックなどをまとめました。ソフトウェアドキュメンテーションで誤りがちな用字・用語や、あいまいな表現方法などを紹介し、特に注意すべき点を整理しました。

付録では、開発用ドキュメントのテンプレート集と各記述項目の例、ユーザードキュメントの各記述項目の例を紹介しています。記述内容や記述の仕方は、対象のソフトウェアに

( 中略 )

# 第1章 開発用ドキュメントとは

ソフトウェアを開発するうえで、ドキュメントの重要性は計り知れません。なぜ重要なのかは本書において後のページで解き明かされますが、ソフトウェア開発の現場ではドキュメントについてどのように考え、扱われているのでしょうか？

あなたはどちらの立場ですか？

- (1) ドキュメントはソフトウェアの開発を進めるうえで必要不可欠なものである。工程ごとにきっちりとドキュメントを作成し、開発を進めている。
- (2) ソフトウェアの中心はあくまでプログラムである。ドキュメントはプログラムのための参考情報、補助的な役割を果たす文書に過ぎない。ドキュメントがなくてもプログラムは作成できる。余裕があるときあるいは開発が終わった後にドキュメントを作成している。

特に(2)の立場の人からは、以下のような声がよく聞かれます。

「ソフトウェアを開発する上でドキュメンテーションが重要なことは分かります。でも、それは理想ですよね。現実には厳しい納期があるんです。仕様もすぐ変わるし。つまり設計書などのドキュメントを作っている暇がないんです。」

上記の立場を整理してみましょう。

- ・ 第1の考え：ドキュメントをきちんと作ってソフトウェアを開発する。  
お客様との契約、設計の整理、レビュー、進捗管理、コーディング効率、プログラムのテスト、納入後の変更対応など、開発する上でどれもスムーズに進みます。
- ・ 第2の考え：ドキュメントを作らずにソフトウェアを開発する。  
ドキュメントなど作らなくても、契約はうまくいくし、進捗も管理できます。設計書などなくてもコーディングやデバッグは効率的にできます。納入後の設計変更にもドキュメントなしですぐに対応できます。

さて、どちらが正しいのでしょうか？どちらがあるべき姿でしょうか？どちらも一理ありそうです。しかし、「理想は分かるが現実的でない」「現実はそのだがそれでは困る」・・・どうすればよいのでしょうか？

そもそも、「ソフトウェア開発」とは何かということ突き詰めて考えると、決してプログラムを作ることを意味するのではないということが分かります。むしろ、ドキュメンテーションこそがソフトウェア開発の中核を成しているといってもよいでしょう。JIS規格 (X0126「応用システムの文書化要領」) においても、「どんなソフトウェア開発プロジェクトであってもそれを成功させるためには、文書化は本質的な役割を果たしている」と述べられています。

文書化とはつまりドキュメンテーションのことです。それでは、なぜそう言えるのか、この章で考えていきましょう。

( 中略 )

## 第3章 外部仕様書

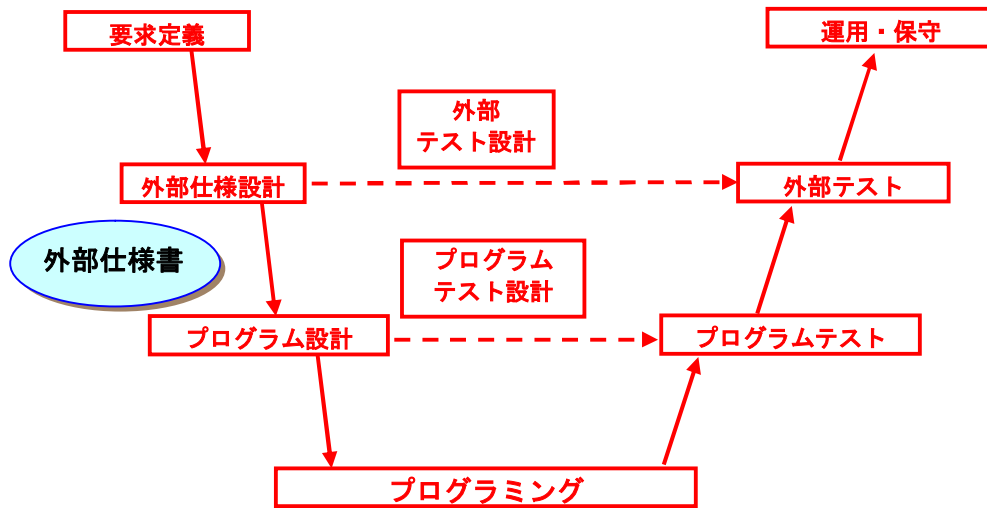


図 3-1:開発工程における外部仕様書の位置づけ

外部仕様書には、「このようなものをこのような作り方で作りたい」といった内容を記述します。

### 3.1 外部仕様書の役割

外部仕様書は外部仕様設計工程の作業結果として出力されるべきドキュメントです。外部仕様書の作成は外部仕様設計そのものであると考えてよいでしょう。そして、外部仕様設計の対象単位に対しての要求に応えるために、どのような外部条件と制約条件のもとで、どのような機能を、どのような使用方法で提供するかを明確にします。つまり、外部仕様書は要求仕様書に示された要求を受けて、開発者側としてどのようなプログラムを提供するかを約束するドキュメントです。

また、外部仕様書では、プログラムの条件・機能・使用方法をより具体化し、次のプログラム設計工程での「どう構築するか」を考える内部作業につなげていきます。

### 3.2 要求仕様書、プログラム設計書との関係

要求仕様書は、要求者の立場で書いた仕様書です。それに対し、外部仕様書は開発者の立場で書く仕様書です。まだあいまいさが残る要求仕様書を元に、開発者側としてどのようなプログラムを提供するのか、その仕様を明確に定義します。

そして、外部仕様書では作るべきプログラムの外観を記述します。つまり、外から見たプログラムの振る舞いを定義します。プログラム内部の作り方には、原則として踏み込みま

( 中略 )

# 第11章 ドキュメンテーションの作法とは

第一部、第二部では、ソフトウェアドキュメントの種類や役割などを解説しました。第三部では、分かりやすいドキュメントを作成するには何に気をつけるべきかについて考えます。ソフトウェアドキュメンテーションにおいて、誤りやすい内容を集め、ライティングのテクニックをまとめます。そして、本章ではそれらのテクニックをドキュメンテーションの作法として品質を高める要素と位置づけ、解説します。

## 11.1 分かりやすさとは

ソフトウェアドキュメンテーションにおいて、分かりやすさとはどのようなことでしょうか？

第一部・第二部で述べたようにソフトウェアドキュメントには、さまざまな種類があります。各ドキュメントにはそれぞれの役割がありました。そして、種類や役割の違いにかかわらず、ソフトウェアドキュメントすべてに共通した役割があります。それは、「読み手に情報を正確に伝える」ことです。伝える内容には、書き手の個人的な主観や予測が入ってはいけません。客観的かつ論理的に記述する必要があります。

つまり、ソフトウェアドキュメントとして何よりも大切なのは、読み手に情報を伝えるだけでなく、読み手にその情報の内容を理解してもらうことなのです。なぜなら、すべてのソフトウェアドキュメントは、ドキュメントを読んだ後で読み手がなんらかのアクションを起こすことを目的としているからです。開発用ドキュメントは、開発の手段となります。ユーザードキュメントは、ユーザーがソフトウェアを理解したり、使いこなしたりするためのものです。そして読み手に理解させるためには、分かりやすいドキュメントでなければなりません。分かりやすさこそが、ソフトウェアドキュメントの品質です。

では、分かりやすいドキュメントとは、具体的にはどういったものなのでしょうか？

まず、文法的に正しい文で記述されていなければなりません。これは、もっとも基本的な要件であり大前提です。文法的に正しい記述でなければ、読み手に伝えたい情報を正しく理解してもらうことが期待できなくなります。そして、文法的に正しいだけでなく表現方法にも気を配らなければなりません。表現方法としては、読みやすさや明確さ、簡潔さなどに注意する必要があります。次章以降では、文法や表現方法についてのポイントを解説します。

## 11.2 分かりにくい記述例

まずは、ソフトウェアドキュメンテーションにおいて分かりにくい記述にはどんなものがあるのか考えてみましょう。いずれも目にしたことがある記述かもしれません。なぜ分かりにくいのか説明できますか？ 以下の問題を解いてみてください。

**次のうち、品詞の使い方として不適切な箇所を正してください。**

(問 1) 再設定を行います。

( 中略 )

**次のうち、構文および係り受けとして不適切な箇所を正してください。**

(問 7) プログラムの品質が、ドキュメントを残すことにより上がります。

( 中略 )

**読みやすく書き替えてください。**

(問 14) このボタンにより設定ダイアログが表示される。設定ダイアログは、シミュレーションのために用意されています。

( 中略 )

**なぜ分かりにくいのでしょうか？**

(問 16) 青い操作ボタンと切り替えスイッチがあります。

( 中略 )

**簡潔な表現に書き替えてください。**

(問 26) ホームページから有効なデータをダウンロードできますが、データの中にウイルスが入り込んでいる危険性があるため、使用前には必ずウイルスチェッカーを実行してください。

( 中略 )

**次のうち、不適切な箇所があれば正してください。**

(問 29) ダイアログを表示したら、設定したい値を入力してください。

( 中略 )

**不要な語句を省いてください。**

(問 37) 在庫があるため、出荷における影響はありません。

( 中略 )

いくつ答えられましたか？ それぞれの内容について次章以降で解説します。

次章では正しい文法を再確認し、そのうえで第 13 章では分かりやすい表現のテクニックを身につけてください。

そして、ぜひ皆さんや同僚が作成しているドキュメントについてもチェックしてみてください。今まで見えなかったものが見えてくるかもしれません。

( 中略 )

# 付録A 開発用ドキュメントのテンプレート集

第一部で述べた各開発用ドキュメントの記述項目を基に、典型的なテンプレートを用意しました。各ドキュメントの目次作成のヒントとしてお使いください。また、テンプレート内の各記述項目には、本文で述べた要点や注意点を挙げています。記述項目に対して、何を書くべきかを整理するきっかけとしてください。書くべき項目の詳細な解説が必要なおき、本文に戻っての内容確認が容易なように、本文中のページと記述項目名も示しました。

しかし、このテンプレートがどんなプログラムのどんな開発用ドキュメントにも適用できるわけではありません。記述項目名は、記述対象のドキュメントの目次タイトルとして適さない場合は、適宜タイトルの検討が必要です。また、ドキュメントに合わせて、記述項目とその内容を取捨選択し、必要に応じて、項目の追加も必要となることを忘れないでください。

## A.1 要求仕様書のテンプレート

### 1. はじめに

#### 1.1 本書の目的

p12 ●本書の目的

- ・記述対象のドキュメントが、開発するプログラムに対して、どのような役割を担っているかを記述する。

#### 1.2 関連文書

p12 ●関連文書

##### (1)上位文書

- ・要求仕様書は、ソフトウェア開発工程においてスタートに位置するドキュメントである。しかし、要求者が基にしている企画書や提案書などのドキュメントが、要求仕様書を作成する上で存在する場合に、上位文書として記載する。存在しなければ、本項目は必要ない。

##### (2)参照文書

- ・必要に応じて参照するドキュメントの名称とバージョンを記載する。

#### 1.3 表記法

p12 ●表記法

- ・必要に応じて記載する。

### 2. 開発目的

p17 ●開発目的

- ・どのような課題の解決を目的としているかを記述する。

### 3. プログラムの用途と使用者

p19 ●用途●使用者

- ・どのようなところ、どのような場合に、どのようなユーザーがプログラムを使用するかを記述する。

(次ページに続く)

( 以下省略 )